In the Claims

1. **(Currently Amended)** A processor-readable medium comprising processor-executable instructions for:

parsing an input file to recognize a file format of the input file, wherein the parsing repeatedly parses <u>once</u> with <u>each of</u> a plurality of component parsers contained within a compound parser, wherein ~~the compound parser is extensible and~~ each of the plurality of component parsers is configured for recognition of a specific file format <u>by which an input file is configured, wherein the compound parser is extensible, and wherein extending the compound parser comprises adding an additional component parser</u>;

checking contents of the input file, according to the recognized file format, to determine whether executable code exists within the input file<u>, wherein the checking comprises detecting executable code because its location within the input file is inconsistent with the recognized file format</u>;

continuing to parse the input file <u>until a component parser recognizes the file format of the input file</u> ~~with all remaining component parsers after at least one component parser recognizes the file format of the input file~~; and

sending a status in response to results of said checking<u>, wherein sending a status comprises further instructions for</u>:

sending a file-has-no-code status when the file format of the input

file was recognized and no executable code was found;

sending a file-has-code status when executable code was found; and

sending a don't-know status when the file format of the input file

was not recognized.

2-5.   (Cancelled)

6.   (Original) The processor-readable medium as recited in claim 1, wherein sending the status comprises further instructions for sending the status to an email program.

7.   (Original) The processor-readable medium as recited in claim 1, wherein sending the status comprises further instructions for sending the status to an instant messaging program.

8.   (Original) The processor-readable medium as recited in claim 1, wherein sending the status comprises further instructions for sending the status to an internet browsing program.

9-12. (Cancelled)

13.   (Currently Amended) The processor-readable medium as recited in claim 11, additionally comprising further instructions for continuing to parse the

input file with all remaining component parsers after at least one

component parser recognizes the file format of the input file ~~parsing the~~

~~input file until a component parser recognizes the file format of the input~~

~~file.~~

14.   **(Currently Amended)** A method of detecting code-free files, comprising:

identifying a new file format, wherein ability to recognize the new file

format is functionality to be extended to a compound parser;

configuring a new component parser according to the new file format,

wherein the new component parser is configured to recognize files

of the new format and also to recognize executable code in files of

the new format by locating executable code that is inconsistent with

the new file format; and

extending functionality of the compound parser by adding the new

component parser to the compound parser;

wherein the compound parser, having extended functionality, is configured

to operate to parse an input file by:

parsing ~~an~~ the input file with ~~a~~ the compound parser, wherein the

compound parser is configured to include a plurality of

component parsers, wherein each component parser is

configured to recognize a specific data file format;

analyzing contents of the input file according to the recognized

specific file format, where available, to determine if the input

file contains executable code; and

sending a status in response to results of said analyzing.

15. **(Original)** The method as recited in claim 14, additionally comprising:

sending a file-has-no-code status when the file format of the input file was

recognized and no executable code was found; and

sending a file-has-code status when executable code was found.

16. **(Original)** The method as recited in claim 14, additionally comprising

sending a don't-know status when a file format of the input file was not

recognized.

17. **(Original)** The method as recited in claim 14, additionally comprising

sending the status to an email program.

18. **(Original)** The method as recited in claim 14, additionally comprising

sending the status to an instant messaging program.

19. **(Original)** The method as recited in claim 14, additionally comprising

sending the status to an internet browsing program.

20. **(Original)** The method as recited in claim 14, wherein parsing the input file

comprises parsing the input file with each of the plurality of component

parsers within the compound parser.

21.   **(Currently Amended)** An apparatus for detecting code-free files, comprising:

a compound parser configured to repeatedly parse an input file, wherein each component parser within the compound parser is configured to recognize executable code within a specific file format selected from among a group of data file formats; and

a controller to examine success of each of the component parsers to recognize the specific file format for which it was configured to recognize and to find executable code within the input file, wherein the controller is configured to send a status in response to results of said checking, wherein sending a status comprises:

> sending a file-has-no-code status when the file format of the input file was recognized and no executable code was found;
>
> sending a file-has-code status when executable code was found; and
>
> sending a don't-know status when the file format of the input file was not recognized.

22.   **(Cancelled)**

23.   **(Original)** The apparatus as recited in claim 21, wherein the apparatus for detecting code-free files is additionally configured to send the status to an email program.

24. **(Original)** The apparatus as recited in claim 21, wherein the apparatus for detecting code-free files is additionally configured to send the status to an instant messaging program.

25. **(Original)** The apparatus as recited in claim 21, wherein the apparatus for detecting code-free files is additionally configured to send the status to an internet browsing program.

26. **(Original)** The apparatus as recited in claim 21, additionally configured to send the status to:

a firewall;

a host intrusion detector; or

a host vulnerability assessor.

27. **(Original)** The apparatus as recited in claim 21, additionally configured to send the status to a program selected from a group of programs, comprising:

a backup program;

a CD/DVD burning program; and

a P2P file-sharing program.

28.   **(Original)** The apparatus as recited in claim 21, wherein each of the component parsers is configured to recognize one of a plurality of data file formats.

29.   **(Original)** The apparatus as recited in claim 21, wherein the compound parser is configured to allow extension by addition of a new component parser to the compound parser, wherein the new component parser recognizes a further file format and recognizes executable code within the further file format.

30.   **(New)** The processor-readable medium as recited in claim 1, wherein adding an additional component parser comprises instructions for:

identifying a new file format, wherein ability to recognize the new file format is functionality to be extended to the compound parser;

configuring a new component parser according to the new file format, wherein the new component parser is configured to recognize files of the new format and also to recognize executable code in files of the new format by locating executable code that is inconsistent with the new file format; and

extending functionality of the compound parser by adding the new component parser to the compound parser.